Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Kierunek: **Informatyka Techniczna (ITE)**
Specjalność: **Inżynieria Systemów Informatycznych (INS)**

PRACA DYPLOMOWA MAGISTERSKA

Wykorzystanie algorytmów genetycznych w systemach wykrywania intruzów w sieciach komputerowych

inż. Bartosz Błyszcz

Opiekun pracy
dr inż. Tomasz Babczyński

Słowa kluczowe: 3-6 słów

WROCŁAW, 2024

Streszczenie

Wykaz skrótów

Tabela 1. Tabela skrótów
Źródło: opracowanie własne

GA	*Genetic Algorithm*	Algorytm Genetyczny
GP	*Genetic Programming*	Programowanie Genetyczne
GNB	*Gaussian Naive Bayes*	Naiwny Klasyfikator Bayesa wykorzystujący rozkład Gaussa
NNs	*Neural Network*	Sieć neuronowa
ML	*Machine Learning*	Uczenie maszynowe
SVM	*Support Vector Machine*	Maszyna Wektorów Nośnych

Spis treści

1. Wstęp

1.1. Wprowadzenie i uzasadnienie tematu pracy

Klasyfikacja danych tabelarycznych jest zagadnieniem, które na codzień dostarcza wyzwań jej twórcom z powodu mnogości danych, a także mnogości cech, a także z nierzadko małą ilością próbek. Jednym z problemów jest między innymi dobór odpowiedniego algorytmu do problemu. Dane tebalryczne występują w każdej dziedzinie, przez co raz na jakiś czas proponowane są nowe rozwiązania i algorytmy mające rozwiązać problem klasyfikacji w sposób lepszy i wydajny. Częśc twórców próbuje podchodzić do tego w sposób innowacyjny, lecz nie zawsze to wychodzi z powodu chociażby dosotoswania algorytmu pod konkretną strukturę danych, co powoduje problemy z wykorzystaniem rozwiązania dla innych danych.

Obecnie jednymi z najpopularniejszych algorytmów do klasyfikacji danych są logiczna regresja(*ang. logistic regression*), drzewo decyzyjne(*ang. decision tree*), losowy las(*ang. random forest*), maszyna wektorów nośnych(*ang. support vector machine*), naiwny bayes(*ang. Naive Bayes*). Dlatego też bardzo ważne jest porównanie wytworzonego wcześniej rozwiązania z grupą innych algorymów, które próbują przetworzyć ten sam zestaw danych.

W dzisiejszych czasach próba taka jest bardzo uproszczona chociażby przez takie platformy jak *Machine Learning Studio*, które pozwalają na wykorzystanie mocy obliczeniowej sklasteryzowanych jednostek wirtualnych do wykonywania obliczeń na odpowiednich maszynach wirtualnych, a także do budowania skomplikowanych zautomatyzowanych procesów złożonych z wielu zadań(*ang. pipeline*). W związku z czym możliwośc wykorzystania platformy chmurowej pozwoli na zautomatyzowanie procesu porównawczego oraz oddelegowanie zadań od chmury obliczeniowej co pozwoli na uniezależnienie powodzenia doświadczenia od mocy obliczeniowej komputera lokalnego, a także na ukazanie całościowo procesu porównania algorytmów klasyfikacyjnych.

1.2. Cel pracy dyplomowej

Celem niniejszej pracy dyplomowej jest porównanie algorytmu klasyfikacji danych tabelarycznych wypracowanego w trakcie pisania pracy inżynierskiej, do algorytmów dostępnych w aplikacji *Machine Learning Studio* znajdującej się na platformie *Microsoft Azure*.

1.3. Założenie techniczne

Dane prezentowane w **tabeli** 1.1 określają podstawowe założenia techniczne przyjęte w trakcie wykonywania analizy porównawczej. Dane te dotyczą między innymi środowiska, w którym wykonane było doświadczenie. Dodatkowo uwzględniono zestaw danych oraz biblioteki użyte w trakcie tworzenia doświadczenia.

Tabela 1.1. Założenia techniczne pracy dyplomowej
Źródło: Opracowanie własne

Środowisko uruchomieniowe	Machine Learning Studio[1]
Język oporogramowania	Python 3.x
Wykorzystane biblioteki	scikit-learn [**sckit-learn**]
	Numpy [2]
	Pandas [3, 4]
Wykorzystane dane	CICDS2017 [5]

2. Sieci neuronowe

3. Klasyfikacja danych

4. Podejście Low-Code

5. Microsoft Azure

6. Opis doświadczenia

7. Analiza porównawcza

8. Perspektywy rozwoju

Wykaz rysunków

Wykaz tabel

Bibliografia

[1] Microsoft. „*Microsoft Machine Learning Studio (classic)*". URL: *https://studio.azureml.net/*.

[2] Charles R Harris i in. „*Array programming with NumPy*". W: *Nature* 584 (7824 wrz. 2019), s. 356–362. DOI: *9.1038/s41586-020-2649-2*.

[3] The pandas development team. „*pandas-dev/pandas: Pandas*". Lut. 2019. DOI: *9.5281/zenodo.3509134*. URL: *https://doi.org/9.5281/zenodo.3509134*.

[4] Wes McKinney. „*Data Structures for Statistical Computing in Python*". W: 2010, s. 56–61. DOI: *10.25080/Majora-92bf1922-00a*.

[5] UNB. „*CICIDS2017 | Kaggle*". URL: *https://www.kaggle.com/datasets/cicdataset/cicids2017*.